**Artificial and Computational Intelligence Assignment 2**

|  |
| --- |
| **ACI Group 189** |
|  |
| **NAMES & BITS ID :** |
| **SHIVAM SAHIL - 2023AA05663** |
| **GIRIJA SHANKAR SAHOO - 2023AA05235** |
| **SOURAJEET SAHOO - 2023AA05029** |
| **MANO RANJAN SAHU - 2023AA05738** |
| **S SRIDHAR - 2023AB05096** |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question 1:**

The Minimax algorithm can be adapted to a two-player solution-based crossword puzzle where points are awarded based on word length. Here's a breakdown of how it would work:

**Players:**

* Maximizer: This represents the player who wants to score the highest points.
* Minimizer: This represents the opponent who wants to minimize the Maximizer's score.

**Game State:**

* The current state of the crossword puzzle with filled with # and blank squares.
* List of available words for the current player.

**Score:**

* Points awarded for a placed word based on its length.
* Penalty point for placing a word incorrectly (-1).

**Minimax Algorithm:**

1. **Base Case:** If the puzzle is solved (all squares filled) or no more words are available, calculate the score based on filled words (Maximizer's points) and any penalty points. This is the final score for this branch of the game tree.
2. **Maximizer's Turn:** For each available word:
   * Simulate placing the word on the board (considering valid crossword rules).
   * Recursively call Minimax for the Minimizer's turn with the updated board state and remaining words (excluding the placed word).
   * Get the score returned by the Minimizer.
   * Choose the word placement that leads to the **highest score** returned by the Minimizer calls (Minimizer tries to minimize Maximizer's score, so the Maximizer picks the move that forces the Minimizer's worst outcome).
3. **Minimizer's Turn:** Similar to Maximizer's turn but with the opposite goal. Here, the Minimizer chooses the word placement that leads to the **lowest score** for the Maximizer (returned by the recursive Minimax calls).

**Recursive Search:**

* Apply the Minimax algorithm recursively to explore possible future game states resulting from each player's actions. This process continues until a certain depth or until a terminal state is reached (e.g., the puzzle is completed)

**Alpha-Beta Pruning:**

* To improve efficiency, implement alpha-beta pruning to prune branches of the search tree that are known to be irrelevant.

**Choose the Best Move:**

* After evaluating all possible word placements, the Maximizer (current player) chooses the word that leads to the **highest score** based on the results from the Minimax exploration.

**Apply Move:**

* Update the game state with the chosen move and continue the game until completion.

**Endgame:**

* Determine the winner based on the final game state (e.g., the player with the highest score or the one who completed the puzzle)

**Challenges and Improvements:**

* The naive implementation can be computationally expensive for large puzzles and word lists. Pruning techniques like Alpha-Beta pruning can be used to eliminate branches that cannot possibly lead to the best outcome.
* A simple score based on word length might not be ideal. Heuristics can be incorporated to consider factors like bonus points for specific word placements or penalties for unused high-value words.

**Question 2:**

* The problem is to predict the most suitable water resource for a given location.
* The decision tree in the problem statement shows the different factors that can be used to make this prediction, such as the distance to the nearest lake, river, and beach, the average monthly rainfall, and whether the location has a sandy aquifer.
* The decision tree is a graphical representation of the decision-making process. It starts with a root node i.e. Lake Distance, which represents the initial decision. The tree then branches out into different nodes, each representing a possible outcome of the decision. The final nodes in the tree represent the different water resources like lake, river, ground water or rain.
* To use the decision tree to make a prediction, we start at the root node and follow the branches that correspond to the given attributes. The path we take will eventually lead us to a leaf node, which represents the most likely water resource for the location.

To use these rules to predict the water resource for a given location, we can use the following procedure:

1. Ask the user to provide the values for the lake distance, river distance, rainfall intensity, sandy aquifer, and beach distance attributes.
2. Use the rules to determine the most likely water resource for the location.

**Examples:**

Here is an example of how the decision tree can be used to make a prediction:

Given attributes:

Lake distance: 12 km

River distance: 10 km

Rainfall intensity : 200 mm

Sandy aquifer: No

Beach distance: 8 km

Path through the decision tree:

Distance to the nearest lake is greater than 10 km.

Distance to the nearest river is greater than 8 km.

Rainfall intensity is greater than 150 mm.

Leaf node: Rain

Prediction: The most likely water resource for this location is Rain.

**Scenario 1:**

Lake distance: 9 km

River distance: 10 km

Rainfall intensity : 200 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is Lake.

A screenshot of a computer

Description automatically generated

**Scenario 2:**

Lake distance: 12 km

River distance: 6 km

Rainfall intensity : 190 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is River.

A screenshot of a computer

Description automatically generated

**Scenario 3:**

Lake distance: 12 km

River distance: 6 km

Rainfall intensity : 200 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is Rain.

A screenshot of a computer

Description automatically generated

**Scenario 4:**

Lake distance: 12 km

River distance: 10 km

Rainfall intensity : 160 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is Rain.

A screenshot of a computer

Description automatically generated

**Scenario 5:**

Lake distance: 14 km

River distance: 8 km

Rainfall intensity : 140 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is Rain.

A screenshot of a computer

Description automatically generated

**Scenario 6:**

Lake distance: 12 km

River distance: 8 km

Rainfall intensity : 140 mm

Sandy aquifer: No

Beach distance: 8 km

Prediction: The most likely water resource for this location is Lake.

A screenshot of a computer

Description automatically generated

**Scenario 7:**

Lake distance: 12 km

River distance: 10 km

Rainfall intensity : 140 mm

Sandy aquifer: Yes

Beach distance: 8 km

Prediction: The most likely water resource for this location is Ground water.

A screenshot of a computer

Description automatically generated

**Scenario 8:**

Lake distance: 12 km

River distance: 20 km

Rainfall intensity : 140 mm

Sandy aquifer: Yes

Beach distance: 4 km

Prediction: The most likely water resource for this location is Rain.

A screenshot of a computer

Description automatically generated

**Scenario 9:**

Lake distance: 12 km

River distance: 18 km

Rainfall intensity : 140 mm

Sandy aquifer: Yes

Beach distance: 4 km

Prediction: The most likely water resource for this location is River.

